

Amendments to the Specification:

Please replace paragraph [0003] with the following replacement paragraph:

[0003] While a software project is being developed, certain parameters must be known regarding the use of the resulting software project in order for the project system to provide the correct tools and properties. For example, consider the case in which a software project is developed for use with a particular target architecture or platform. The target architecture or platform which the software project is intended to be used with affects many facets of the software project. For example, one architecture currently in use is a 64-bit® architecture known as IA-64®. If the project is being developed for use on an IA-64 platform, there may be a specific IA-64® programming stylesheet which should be followed for the project. This stylesheet should be accessible to the project system so the project system can guide the developer in developing code according to the stylesheet's dictates. Additionally, the developer can use certain tools or toolsets specific to the IA-64® (or to a set of platforms including the IA-64®). For example, the debugger used to debug the project while in development will be specific to the IA-64®. Additionally, tools for running the code in the IA-64 environment may be required by the developer.

Please replace paragraph [0006] with the following replacement paragraph:

[0006] One way to reduce reduplication of effort is to use templates, for example, enterprise templates as used in MICROSOFT VISUAL STUDIO .NET®. Software architects frequently evaluate new technologies, explore possible ways to use them, and recommend best practices. Through the use of enterprise templates, software architects can provide guidance to development teams about how to work within an architecture by eliminating many decisions involving component and item choices that should not be part of the developer's burden. Enterprise templates simplifies these choices by dynamically modifying certain integrated development environment (IDE) features in VISUAL STUDIO NET®. Enterprise templates

can provide certain beginning project structures and continuous development guidance as components are added, properties exposed, and values assigned.

Please replace paragraph [0036] with the following replacement paragraph:

[0036] When a method of the IVsHierarchy interface is called by an action of a developer using the base project object 200, in the absence of an added flavor, the interface used would be IVsHierarchy interface 210(a). However, client flavor object 220 is being used with the base project object 200. Client flavor 220 is also a COM object, and the client flavor object 220 and the base project object 200 have been aggregated in such a way so as to allow the client flavor object 220 to override or enhance the capabilities of the base project object 200. For example, as shown in FIG. 2, client flavor object 220 implements its own IVsHierarchy interface, IVsHierarchy 230(a). When an IVsHierarchy interface is needed in the IDE where the client flavor object 220 has been applied to base project object 200 as shown in FIG. 2, the IVsHierarchy interface 230(a) will be called first. The same is true for IOleCommandTarget/IVsUIHierarchy interface 230(b) which will be provided to the client rather than IOleCommandTarget/IVsUIHierarchy interface 210(b). Thus, a flavor object can override a base project object. If only part of the interface is implemented in the flavor, the call to the interface can be passed in to the base project object 220. In this way, the flavor can enhance an interface implemented in the base project object 200 without needing to re-implement the entire interface.

Please replace paragraph [0055] with the following replacement paragraph:

[0055] FIG. 4 is a block diagram of extensibility through an interface in a flavor according to one embodiment of the invention. As shown in FIG. 4, an IVsFlavorCfgProvider interface 430(a) is provided on a flavor object 420 which has been applied to a base project object 400. The base project 400 implements a delegate object, base project configuration object ProjectCfg object 440. Project configuration objects allow properties other than the certain properties (common properties) of the project to be made available. Project configuration objects add various configuration properties specific to one kind of build for one platform.

Thus, these configuration-dependent properties are defined in each project configuration object. These project properties can be used to determine, for example, which project items will be included in a particular build, what output files will be created, where the output files will be put, and how they will be optimized.

Please replace paragraph [0056] with the following replacement paragraph:

[0056] In order to allow the flavor to also affect the properties which are defined in the project configuration object, the IVsFlavorCfgProvider interface 430(a) is used to provide a flavor-specific project configuration object FlavorProjectCfg object 450. The base project configuration object delegates to this FlavorProjectCfg object 450, allowing the flavor object 420 to extend the ProjectCfg object 440. The IVsFlavorCfgProvider interface 430(a) functions as a factory for project flavor's configuration objects. The base project system asks the project flavor to create an IVsFlavorCfg object 450 corresponding to each one of its configuration objects. The FlavorCfg object(s) 450 can implement an interface (IPersistXMLFragment interface 455(b)) which manages persistence into the project file.